Parallel Algorithms for Multiphysics Adaptive Mesh Refinement

John Bell

Lawrence Berkeley National Laboratory

DD20
San Diego, CA
February 7-11, 2011

PDE-based multiphysics applications

Many PDE-based applications require high-fidelity simulation of multi-scale / multi-physics phenomena

- Combustion
- Astrophysics
- Subsurface flow
- Climate
- Fusion
- Fission

Characteristic of these problem areas is that they couple a number of different physical processes across a range of length and time scales

How can we exploit the structure of these problems in developing simulation methodology

- What are the characteristics of these type of problems
- What might the algorithms look like
- How can we implement them
- How can effectively utilize modern parallel architectures

A combustion example

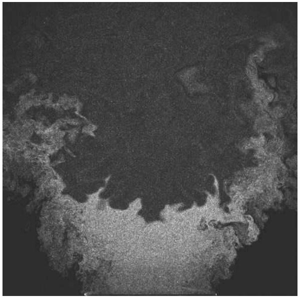
Spatial Scales

- Domain: $\approx$ 10 cm
- Flame thickness: $\delta_T \approx$ 1 mm
- Integral scale: $\ell_t \approx 2 - 6$ mm

Temporal Scales

- Flame speed $O(10^2)$ cm/s
- Mean Flow: $O(10^3)$ cm/s
- Acoustic Speed: $O(10^5)$ cm/s

Fast chemical time scales



Mie Scattering Image

Strategies

- AMR to exploit varying spatial resolution requirements
- Temporal discretization strategies
  - Fully explicit
  - Fully implicit
  - Exploit the multiscale character of the problem

Low Mach number formulation

Exploit separation of scales between fluid motion and acoustic wave propagation

Momentum $$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = -\nabla \pi + \nabla \cdot \tau$$

Species $$\frac{\partial (\rho Y_m)}{\partial t} + \nabla \cdot (\rho U Y_m) = \nabla \cdot (\rho D_m \nabla Y_m) + \dot{\omega}_m$$

Mass $$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0$$

Energy $$\frac{\partial \rho h}{\partial t} + \nabla \cdot \left(\rho h \vec{U}\right) = \nabla \cdot (\lambda \nabla T) + \sum_m \nabla \cdot (\rho h_m D_m \nabla Y_m)$$

Equation of state $p_0 = \rho \mathcal{R} T \sum_m \frac{Y_m}{W_m}$ constrains the evolution

Differentiation of EOS expresses constraint in the form

$$\nabla \cdot U = S$$

where S is a function of the thermodynamic state of the system

Generalized projection formulation

Fractional step scheme

- Advance velocity and thermodynamic variables
  - Specialized advection algorithms
  - Diffusion
  - Stiff reactions
- Project solution back onto constraint – variable coefficient elliptic PDE, multigrid

Stiff kinetics relative to fluid dynamical time scales

$$\frac{\partial(\rho Y_m)}{\partial t} + \nabla \cdot (\rho U Y_m) = \nabla \cdot (\rho D_m \nabla Y_m) + \dot{\omega}_m$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho U h) = \nabla \cdot (\lambda \nabla T) + \sum_m \nabla \cdot (\rho h_m D_m \nabla Y_m)$$

Operator split approach

- Chemistry $\Rightarrow \Delta t/2$
- Advection – Diffusion $\Rightarrow \Delta t$
- Chemistry $\Rightarrow \Delta t/2$

Block-structured AMR

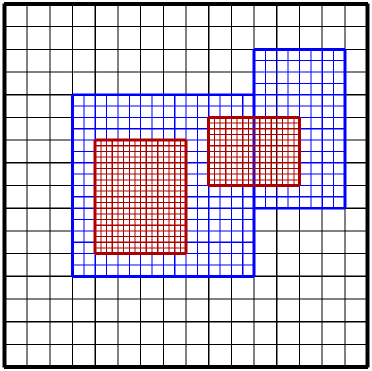
AMR – exploit varying resolution requirements in space and time

Block-structured hierarchical grids

- Amortize irregular work

Each grid patch (2D or 3D)

- Logically rectangular, structured
- Refined in space and (possibly) time by evenly dividing coarse grid cells
- Dynamically created/destroyed


2D adaptive grid hierarchy

- How do we integrate PDE's on this type of grid structure
- How do we implement those algorithms
- How do we parallelize implementations

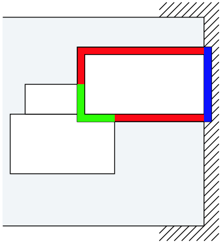
Consider a simple case – Hyperbolic Conservation Laws

AMR for conservation laws

Conservative explicit finite volume scheme

$$U_{i,j}^{n+1} = U_{i,j}^{n} - \frac{\Delta t}{\Delta x}(F_{i+1/2,j} - F_{i-1/2,j}) - \frac{\Delta t}{\Delta y}(G_{i,j+1/2} - G_{i,j-1/2})$$

🟩 Fine-Fine
🟦 Physical BC
🟥 Coarse-Fine



Recursive integration with subcycling in time

- Integrate each grid patch separately
- Fill ghost cells for next finer level, interpolating in space and time from coarser grid where needed
- Integrate fine grid for r time steps

- Berger and Colella, JCP 1989
- Bell, Berger, Saltzman, Welcome, JCP 1994

Coarse and fine grids are at the same time but the overall process isn't conservative.

At c-f edges flux used on the coarse grid and average of fine grid fluxes don't agree

Reflux to make overall integration conservative – update coarse grid with difference in coarse and fine fluxes

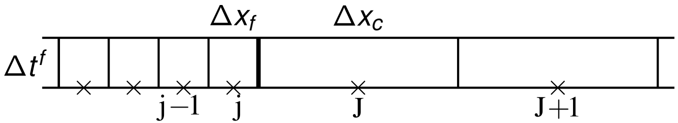
Let's look at this process in more detail

Hyperbolic–1d

Consider $U_t + F_x = 0$ discretized with an explicit finite difference scheme:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = \frac{F_{i-1/2}^{n+\frac{1}{2}} - F_{i+1/2}^{n+\frac{1}{2}}}{\Delta x}$$

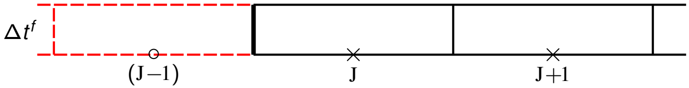
In order to advance the composite solution we must specify how to compute the fluxes:



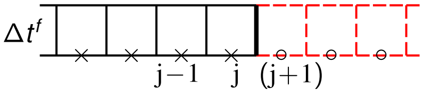
- Away from coarse/fine interface the coarse grid sees the average of fine grid values onto the coarse grid
- Fine grid uses interpolated coarse grid data
- The fine flux is used at the coarse/fine interface

Hyperbolic–composite

One can advance the coarse grid



then advance the fine grid



using “ghost cell data” at the fine level interpolated from the coarse grid data.

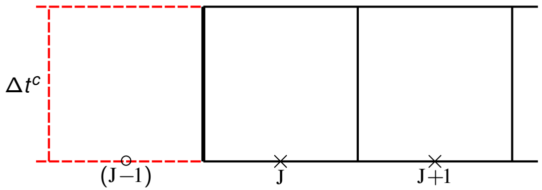
This results in a flux mismatch at the coarse/fine interface, which creates an error in U_J^{n+1}. The error can be corrected by refluxing, i.e. setting

$$\Delta x_c U_J^{n+1} := \Delta x_c U_J^{n+1} - \Delta t^f F^c_{J-1/2} + \Delta t^f F^f_{j+1/2}$$

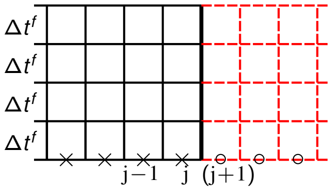
Before the next step average fine grid solution onto coarse grid.

Hyperbolic–subcycling

To subcycle in time we advance the coarse grid with Δt^c



and advance the fine grid multiple times with Δt^f.



The refluxing correction now must be summed over the fine grid time steps:

$$\Delta x_c U_J^{n+1} := \Delta x_c U_J^{n+1} \\ - \Delta t^c F^c_{J-1/2} + \sum \Delta t^f F^f_{j+1/2}$$

AMR Discretization Design

AMR discretization – solve on different levels separately

- Integrate on coarse grid
- Use coarse grid to supply Dirichlet data for fine grid at coarse / fine boundary
- Synchronize to correct errors that arise from advancing grids at different levels separately

Errors take the form of flux mismatches at the coarse/fine interface

Synchronization:

- Define what is meant by the solution on the grid hierarchy
- Identify the errors that result from solving the equations on each level of the hierarchy “independently” (motivated by subcycling in time)
- Solve correction equation(s) to “fix” the solution
- For subcycling, average the correction in time

Elliptic AMR

Look at 1d (degnerate) example

$$-\phi_{xx} = \rho$$

where ρ is a discrete approximation to the derivative of a δ function at the center of the domain

$$\rho^f_J = -\alpha \qquad \rho^f_{J+1} = \alpha$$

but $\rho^c \equiv 0$

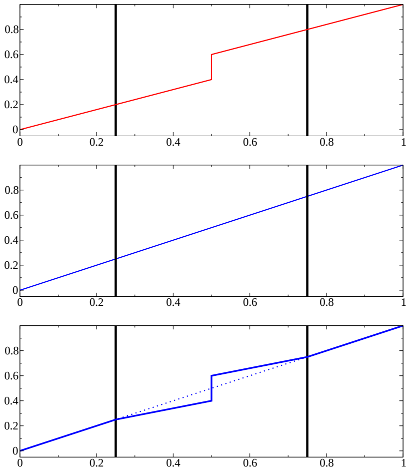
Define a composite discretization

$$L^{c-f}\phi^{c-f} = \rho^{c-f}$$

and solve

Apply design principles above

- Solve $L^c \bar{\phi}^c = \rho^c$
- Solve $L^f \bar{\phi}^f = \rho^f$ using Dirichlet boundary conditions at $c-f$ interface



Elliptic AMR – cont'd

How do we correct the solution

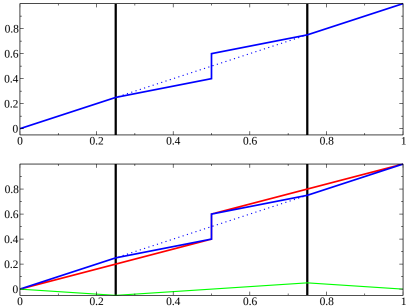
If we define $e = \phi - \bar{\phi}$ then

$$L^{c-f} e = R$$

where $R = 0$ except at $c - f$ boundary where the it is proportional to the jump in ϕ_x.

Solve for e and form $\phi = \bar{\phi} + e$

- e exactly corrects the mismatch
- Residual is localized to the $c - f$ boundary but correction is global
- The error equation is a discrete layer potential problem
- e is a discrete harmonic function on the fine grid $\rightarrow$ solve only on coarse grid and interpolate



Parabolic discretization

Consider $u_t + f_x = \varepsilon u_{xx}$ and the semi-implicit time-advance algorithm:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{f_{i+1/2}^{n+\frac{1}{2}} - f_{i-1/2}^{n+\frac{1}{2}}}{\Delta x} = \frac{\varepsilon}{2}\left((\Delta^h u^{n+1})_i + (\Delta^h u^n)_i\right)$$

The difference e^{n+1} between the exact composite solution u^{n+1} and the solution $\overline{u}^{n+1}$ found by advancing each level separately satisfies

$$(I - \frac{\varepsilon \Delta t^c}{2}\Delta^h)\; e^{n+1} = \frac{\Delta t^c}{\Delta x_c}(\delta f + \delta D)$$

$$\begin{aligned}
\Delta t^c\;\delta f &= -\Delta t^c\;\overline{f}_{J-1/2} + \sum \Delta t^f f_{j+1/2} \\
\Delta t^c\;\delta D &= \frac{\varepsilon \Delta t^c}{2}(\overline{u}_{x,J-1/2}^{c,n} + \overline{u}_{x,J-1/2}^{c,n+1}) \\
&\quad - \sum \frac{\varepsilon \Delta t^f}{2}(\overline{u}_x^{c-f,n} + \overline{u}_x^{c-f,n+1})
\end{aligned}$$

Multiphysics AMR

Basic integration paradigm works for hyperbolic, elliptic and parabolic PDEs

Synchronization equations match the structure of the process being corrected.

Recall combustion algorithm – algorithm components

- Semi-implicit treatment of velocity, enthalpy and species
  - Explicit Advection
  - Crank Nicolson diffusion
- Stiff ODEs
- Elliptic projections

AMR for low Mach number can be constructed by carefully combining the above elements

Key issue is keeping tracking of different aspects of synchronization and performing them in the right order

Same set of tools can be used for a variety of applications

- Incompressible flow
- Self-gravitational compressible flows
- Low Mach number astrophysics
- Porous media flow
- . . .

How can we implement this to be able to reuse the pieces

Implementation

One wants to implement these types of algorithms within a software framework that supports the development of block-structured AMR algorithms

- Represent dynamically changing hierarchical solution
- Manage error estimation and regridding operations
- Orchestrate multistep algorithms and synchronization
- Support for iterative methods for implicit algorithms

There are a number of frameworks that support implementation of these types of algorithms

We use BoxLib

- Data structures
- Operations on those data structures
- Model for parallelization

Data Structures

Index space

- Box : a rectangular region in index space
- BoxArray : a union of Boxes at a level

Real data at a level

- FAB: FORTRAN-compatible data on a single box
  - Data on a patch
  - These patches are quite large – thousands of points
- MultiFAB: FORTRAN-compatible data on a union of rectangles
  - Data at a level
- FluxRegister: FORTRAN-compatible data on the border of a union of rectangles
  - Data for synchronization

Data Operations

Index space Operations:

- Create and manage box topology
- Identify neighbors on same level
- Identify which coarse grids underlie a given fine patch

Single-level operations

- Fill boundary data from same-level grids
- Fill data using physical boundary conditions
- Integrate data at a level
  - Patch by patch for explicit algorithms
  - Solve over all patches at a level for implicit algorithms

Multi-level operations

- Interpolate : coarse $\rightarrow$ fine
- Average : fine $\rightarrow$ coarse
- Fill boundary data from coarser grids
- Synchronization
  - Local corrections for explicit algorithms
  - Implicit synchronization systems for implicit algorithms

Parallel Data Distribution

AMR hierarchy represented by a BoxArray and MultiFAB at each level

- Each processor contains the full BoxArray.
  - Simplifies data-communications: send-and-forget
- Data itself is distributed among processors; different resolutions are distributed independently, separately load-balanced.
- Owner computes rule on FAB data.
- Issues for efficient implementation
  - Dynamic load balancing
  - Efficient manipulation of metadata
  - Optimizing communication patterns
  - Fast linear solvers

Metadata, communications and solvers

Index space operations are naively $O(n^2)$

- Each box needs to know its neighbors
- Bin BoxArray spatially
- Limit searches to boxes in neighboring bins

Communication

- Every MultiFAB with the same BoxArray has the same distribution
- Each processor caches list of its grids' nearest neighbors and their processors
- Each processor caches list of coarse grids and their processors used to supply boundary conditions
- Messages are ganged: no more than one message is ever exchanged between processors in an operation

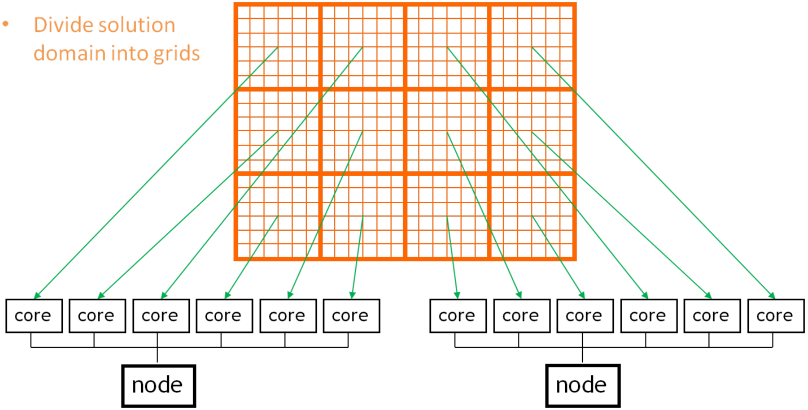
Solvers

- Semi-structured solvers
- Current approaches based on multigrid
  - As problem is coarsened, floating point to communication gets small
  - Communication avoiding algorithms
  - Consolidate data at coarse levels of multigrid

Multi-core architectures

Pure MPI approach

- Divide solution domain into grids

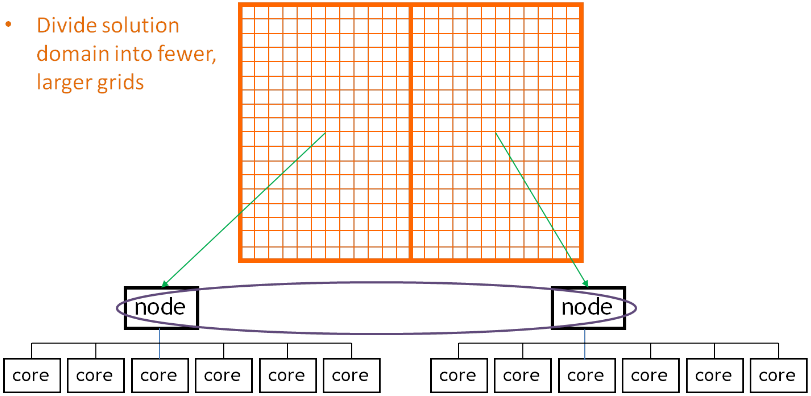


Each grid is assigned to a core

Cores communicate using MPI

Hybrid model

- Divide solution domain into fewer, larger grids



Each grid is assigned to a node

OpenMP used to spawn threads so that cores within a node work on the grids simultaneously

Nodes communicate using MPI

MPI versus Hybrid model

Advantages of hybrid model

- Fewer MPI processes lead to reduced communication time
- Less memory for storing ghost cell information
- Reduced work from larger grids – surface to volume effect

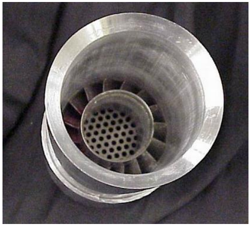
Disadvantages of hybrid model

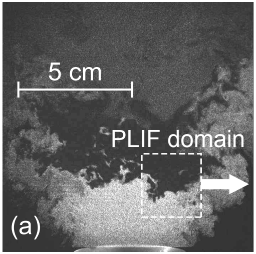
- Spawning threads is expensive – makes performance worse for small core counts
- Can't hide parallelization from physics modules

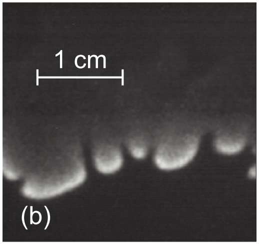
With hybrid model, we have been able to scale multiphysics applications to 100K processors

Hydrogen combustion

DOE's Office of Fossil Energy is interested in developing fuel-flexible turbines that can operate with hydrogen-rich fuels







Detailed simulation is needed to understand the structure of these flames

- OH PLIF shows gaps in the flame
- Standard flame models are not applicable
- Standard experimental diagnostics hard to interpret

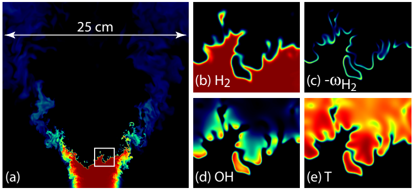
Low swirl burner simulation

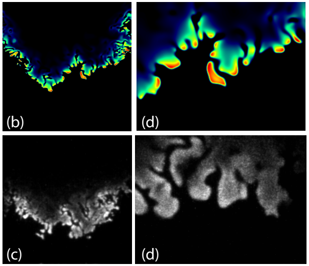
Simulation of lean-premixed hydrogen flame stabilized on a low-swirl burner

- Detailed chemistry and transport
- No explicit models for turbulence or turbulence / chemistry interaction
- 25 cm x 25 cm x 25 cm domain
- Methodology enables simulation at effective resolution of 2048^3

Simulation captures cellular structure of thermodiffusively unstable lean hydrogen flames

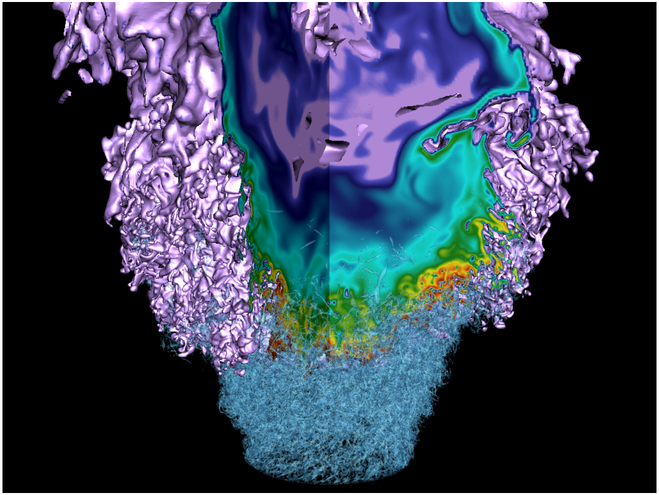
- Quantify enhanced burning from local enrichment of the fuel resulting for high H_2 diffusion
- Provide insight into the analysis of experimental diagnostics





Experiment vs. simulation

Hydrogen low swirl burner



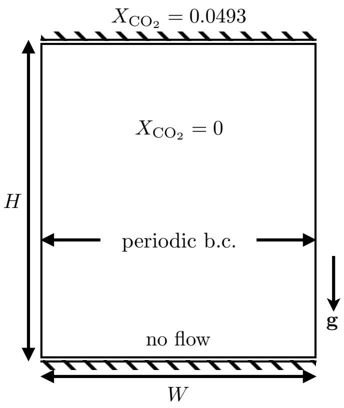
Animation of OH (flame marker) and vorticity

Convection / diffusion in CO_2 sequestration

CO_2 is injected into the subsurface and forms a liquid layer on top of resident brine.

However, when CO_2 dissolves into the brine it increases the density, inducing gravity driven convection

This effect can potentially increase the storage capacity of the formation



Simplified model

Model the system as a two-component incompressible flow with diffusive injection of CO_2 at the top of the formation. Look at dynamics at small scales.

$$\frac{\partial \phi \rho X_\alpha}{\partial t} + \nabla \cdot (\rho X_\alpha v_T) = \nabla \cdot \phi \tau \mathcal{D} \rho \nabla X_\alpha, \quad \alpha = 1, 2$$

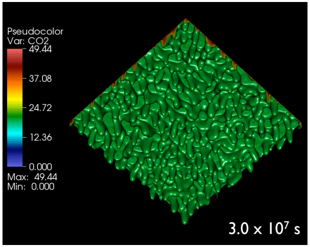
Augemented with an equation of state

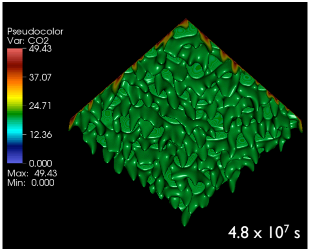
$$\rho \left(\frac{X_1}{\rho_1} + \frac{X_2}{\rho_2} \right) = 1$$

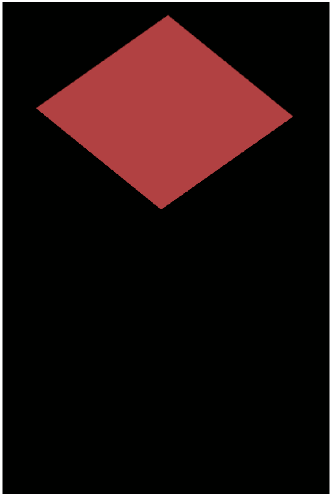
For this system diffusion is important and effects the divergence of the total velocity

$$\nabla \cdot v_T = \sum_{\alpha=1}^{2} \frac{1}{\rho_\alpha} \nabla \cdot \phi \rho \tau \mathcal{D} \nabla X_\alpha$$

3D simulation







Summary and conclusions

Multiphysics applications characterized by a wide range of length and time scales

Approach to developing simulation methodology

- Analysis of relationship of temporal scales
- Mathematical formulation that exploits those relationships
- Numerics for each process that reflects character of the process
- Block-structured AMR for spatially varying resolution requirements
  - How to integrate PDEs in AMR grids
  - Framework for implementation of AMR algorithms
  - Structured grid AMR provides natural model for hierarchical parallelism